Integer Affine Transformations of Parametric $\mathbb{Z}$-polytopes and Applications to Loop Nest Optimization

RACHID SEGHIR
University of Batna, Algeria

VINCENT LOECHNER
INRIA team CAMUS and LSIIT lab, University of Strasbourg, France

BENOÎT MEISTER
Reservoir Labs, New York

November 10, 2010

Abstract

The polyhedral model is a well-known compiler optimization framework for the analysis and transformation of affine loop nests. We present a new method concerning a difficult geometric operation that is raised by this model: the integer affine transformation of parametric $\mathbb{Z}$-polytopes. The result of such a transformation is given by a worst-case exponential union of $\mathbb{Z}$-polytopes. We also propose a polynomial algorithm (for fixed dimension), to count points in arbitrary unions of a fixed number of parametric $\mathbb{Z}$-polytopes. We implemented these algorithms and compared them to other existing algorithms, for a set of applications to loop nest analysis and optimization.
Keywords: Polyhedral model, memory hierarchy optimization, parametric $\mathbb{Z}$-polytopes transformation and enumeration, Ehrhart polynomials, counting solutions to Presburger formulas.

1 Introduction

Many affine loop nests analyses and optimizations raise the problem of counting the number of images by an affine integer transformation of the lattice points contained in a parametric polytope. This problem was raised in this area a twenty years ago [25, 26]. Since then, many solutions have been proposed, but none of them conclusively solves it. The problem of counting solutions to a Presburger formula is equivalent to counting points in the image by integer affine transformations of a union of parametric $\mathbb{Z}$-polytopes. A parametric $\mathbb{Z}$-polytope

is the intersection between a parametric polytope (a bounded polyhedron, defined by a set of constraints depending linearly on formal parameters) and an integer lattice.

Early solutions compute an approximation of this number [38, 29]. Some solved the subproblem of counting the integer points in a polytope (or equivalently in a single $\mathbb{Z}$-polytope), depending on one [2, 10] or many [7, 35, 36, 1] parameters. Then, some exact methods to count integer affine images of $\mathbb{Z}$-polytopes were proposed for the non-parametric case [24, 5, 39]. Some applications were developed without solving the general-case problem, but many of them did not reach optimality [38, 29], or were restricted as they did not support formal parameters [39]. Other applications need the exact solution [21, 32].

Pugh [26] first proposed an algorithm to solve the general-case problem, based on the Fourier-Motzkin variable elimination [9]. However, it seems that no implementation is available, probably due to the complexity of this method: the number of *splinters* which have to be built is an exponential function of the coefficients of the formula. Moreover, it is not clear whether it is able to eliminate more than one existential variable without scanning an exponentially large number of polytopes.

Verdoolaege et al. [34] proposed to apply simple rewriting rules (removing existential variables that are *unique* or *redundant*, decomposition in *independent splits*) to a disjoint union of parametric sets computed using the Omega library [25]. If these rewriting rules fail, they solve a parametric integer linear programming problem (using the PIP library [13]), like Boulet and Redon in [6]. But the input of their algorithm, the disjoint union of sets computed from a Presburger formula, is worst-case exponential; and PIP is worst-case exponential. They also do not compute the actual integer affine image of the union of parametric $\mathbb{Z}$-polytopes, but count the integer points of an equivalent set: if the application needs not only the number of points, but also the image itself, it cannot be used.

Another algorithm for computing the integer projection of a polytope was first proposed by Barvinok and Woods [3], and then extended to parametric case by Verdoolaege and Woods [37]. Even if this algorithm is *theoretically* polynomial-time (for fixed dimension), we believe that it still remains unusable in practice, because it uses algorithmic flatness theory and iterated Boolean combinations of rational generating functions. Indeed, this algorithm has frequently been referred to as "unimplementable" [17]. A first-ever *preliminary* implementation of Barvinok-Woods's algorithm has been reported by Köppe et al. [17]. But this implementation seems to be rather slow even for simple examples. The implementation itself is considered a challenge.

More recently, Lasaruk and Sturm [19] proposed a framework consisting in a generalization of Presburger arithmetic, where the coefficients are arbitrary polynomials in non-quantified variables. The elimination of the existential variables may result in a large number of atomic formulas function of the parameters (even for small examples). Furthermore, the atomic formulas may still contain bounded quantifiers. In order to expand this result into disjunctions, the parameters have to be instantiated. Hence the final result could not be given in a

symbolic form (as a function of the parameters).

In this paper, we propose a new method handling this problem: (i) a polynomial-time algorithm to count points in the union of a fixed number of parametric $\mathbb{Z}$-polytopes of fixed dimension, (ii) an algorithm to compute the integer affine image of a union of parametric $\mathbb{Z}$-polytopes, in the form of a worst-case exponential union of parametric $\mathbb{Z}$-polytopes. We also compare our implementation with other interesting and recent approaches, and show that it is efficient and can be used to optimize real applications.

We will provide the reader with some references on using our work in compiler design and program optimization. The class of *parametric affine loop nests* (programs based on affine references to arrays in affine bounded loop nests, containing unknown variables at compile time) has been tackled in the past by many researchers, since it occurs frequently and is resource-consuming, in applications like calculation-intensive scientific applications, digital audio/video, graphics, compression/decompression, radar and DSP, etc. In this context, array linearization for hardware design [32], cache access optimization [14, 8, 21, 31], memory size computation [38, 39], and data distribution for NUMA-machines [15], reduce to the problem of computing integer affine images of a union of parametric $\mathbb{Z}$-polytopes and enumerating them. Other applications have been reported in economics and in mathematics, in the context of combinatorics, representation theory, statistics and discrete optimization (see [10] for references).

The paper is organized as follows: after presenting some basic concepts in section 2, we describe our algorithm for computing the integer affine image of a $\mathbb{Z}$-polytope in section 3. Section 4 presents our algorithm for counting points in arbitrary unions of parametric $\mathbb{Z}$-polytopes. In section 5 we provide a comparative study of our implementation with related work and some applications to loop nest optimization. Finally, the conclusions are given in section 6.

2 Basic concepts

In this section we recall some polyhedral and lattice definitions as well as some theoretical results on counting lattice points in parametric polytopes.

Definition 2.1 *A* **rational polyhedron** $P \subset \mathbb{Q}^d$ *is a set of rational* d*-dimensional vectors* $\mathbf{x}$ *defined by linear inequalities*

$$P = \left\{ \mathbf{x} \in \mathbb{Q}^d \mid A\mathbf{x} \geq \mathbf{c} \right\}, \tag{1}$$

with $A \in \mathbb{Z}^{m \times d}$ *and* $\mathbf{c} \in \mathbb{Z}^m$*. The system* $A\mathbf{x} \geq \mathbf{c}$ *may contain pairs of inequalities that are equivalent to an equality (called implicit equalities). In such a case, the polyhedron is said* non-full-dimensional*.*

A bounded rational polyhedron is commonly called a rational polytope*.*

Definition 2.2 *The* **affine hull** *of a set* $S \subset \mathbb{Q}^d$ *is the set* $\{ \lambda_1 x_1 + \cdots + \lambda_k x_k \mid \{ x_1, \ldots x_k \} \subset S, \lambda_i \in \mathbb{Q}, \sum_i \lambda_i = 1 \}$*.*

Definition 2.3 *The* **dimension** *of a rational polyhedron $P \subset \mathbb{Q}^d$ is the dimension of its affine hull. Equivalently, it is equal to the dimension d of the ambient space $\mathbb{Q}^d$ minus the number of linearly independent (implicit) equalities in the system $A\mathbf{x} \geq \mathbf{c}$.*

Definition 2.4 *A* **rational parametric polytope** *$P_{\mathbf{p}}$ with n parameters $\mathbf{p}$ is a set of rational d-dimensional vectors $\mathbf{x}$ defined by linear inequalities on $\mathbf{x}$ and $\mathbf{p}$*

$$P_{\mathbf{p}} = \left\{ \mathbf{x} \in \mathbb{Q}^d \mid A\mathbf{x} \geq B\mathbf{p} + \mathbf{c} \right\}, \tag{2}$$

with $A \in \mathbb{Z}^{m \times d}$, $B \in \mathbb{Z}^{m \times n}$ and $\mathbf{c} \in \mathbb{Z}^m$, and such that for each fixed value $\mathbf{p}_0$ of $\mathbf{p}$, $P_{\mathbf{p}_0}$ defines a (possibly empty) rational polytope in $\mathbb{Q}^d$.

Definition 2.5 *A* **d-dimensional integer lattice** *is a subset of $\mathbb{Z}^d$ defined by integer linear combinations of linearly independent integer vectors, called lattice-generating vectors (or lattice basis), plus an affine part.*

$$L = \left\{ A\mathbf{x} + \mathbf{c} \mid \mathbf{x} \in \mathbb{Z}^d \right\}, \tag{3}$$

where A is a $d \times d$ integer matrix whose column vectors are the generators of the lattice and $\mathbf{c}$ is a constant integer vector. We also denote such a lattice by $\mathcal{L}(A, \mathbf{c})$.

Definition 2.6 *A* **$\mathbb{Z}$-polytope** *is a regular subset of the integer points contained in a rational polytope. It can also be seen as the intersection between a rational polytope and an integer lattice of the same dimension.*

Definition 2.7 *A d-dimensional* **standard $\mathbb{Z}$-polytope** *is the intersection between a rational d-polytope and the* standard *lattice $\mathbb{Z}^d$.*

Definition 2.8 *An* **integer affine transformation** *of a d-dimensional $\mathbb{Z}$-polytope $\mathcal{Z} = P \cap L$ is the transformation of its points by an integer affine function:*

$$\begin{array}{rccc} T: & \mathbb{Z}^d & \rightarrow & \mathbb{Z}^k \\ & \mathbf{x} & \mapsto & A\mathbf{x} + c \end{array} \tag{4}$$

where A is an integer matrix and c an integer vector.

Definition 2.9 *A* **Presburger formula** *is a set of linear (in)equalities linked by the logical operators ($\neg$, $\wedge$, $\vee$), and the universal and existential quantifiers ($\forall$, $\exists$).*

2.1 Ehrhart Theory

Definition 2.10 *A rational* **n-periodic number** *$U(\mathbf{p})$ is a function $\mathbb{Z}^n \rightarrow \mathbb{Q}$, such that there exists a* **period** *$\mathbf{q} = (q_1, \ldots, q_n) \in \mathbb{N}^{+n}$, with $U(\mathbf{p}) = U(\mathbf{p}')$ whenever $p_i \equiv p'_i \mod q_i$, for $1 \leq i \leq n$.*

Periodic numbers can be represented by an n-dimensional lookup-table $U_{\mathbf{p}}$ such that $U(\mathbf{p}) = U_{\mathbf{p}}[p_1 \bmod q_1, \ldots, p_n \bmod q_n]$. Such lookup-tables are typically written as a list or matrix of values enclosed in square brackets. E.g., $U_N = \left[1, \frac{3}{4}\right]_N$ is a 1-periodic number with period $q_1 = 2$; $U_N = 1$ if $N \bmod 2 \equiv 0$ and $U_N = \frac{3}{4}$ if $N \bmod 2 \equiv 1$. A 2-periodic number of period $(2, 3)$ can be represented as $\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{N,M} = \left[\ [a, b, c]_M,\ [d, e, f]_M\ \right]_N$, which is equal to b if $N = 0$ and $M = 1$.

The lookup-table representation of periodic numbers has the advantage that it can be fully simplified, but it is worst case exponential [35, 36]. Other representations avoid representing periodic numbers with an exponential number of coefficients exist, such as integer parts, modulo and fractional representations. E.g., the lookup-table periodic number $U_N = \left[1, \frac{3}{4}\right]_N$ is equivalent to the following functions: $1 - \frac{N}{4} - \frac{1}{2}\left\lceil -\frac{N}{2} \right\rceil$, $1 - \frac{N}{4} + \frac{1}{2}\left\lfloor \frac{N}{2} \right\rfloor$, $1 - \frac{1}{4}(N \bmod 2)$ or $1 - \frac{1}{2}\left\{\frac{N}{2}\right\}$, where $\left\{\frac{N}{2}\right\}$ is the fractional part of $\frac{N}{2}$.

Definition 2.11 *A* **quasi-polynomial** *of degree d in n variables $\mathbf{p} = (p_1, p_2, ..., p_n)$ is a polynomial expression of degree d in $\mathbf{p}$ over the rational n-periodic numbers*

$$\sum_{\mathbf{i} \in I} U_{\mathbf{i}}(\mathbf{p})\, \mathbf{p}^{\mathbf{i}}, \tag{5}$$

with $I \subset \mathbb{N}^n$, $i = (i_1, ..., i_n)$, $\sum_{k=1}^{n} i_k \leq d$, $U_{\mathbf{i}}$ periodic numbers and $\mathbf{p}^{\mathbf{i}} = p_1^{i_1} p_2^{i_2} ... p_n^{i_n}$. The **period** *of a quasi-polynomial is the componentwise least common multiple (lcm) of the periods of its coefficients $U_{\mathbf{i}}$.*

Definition 2.12 *The* **enumerator** *$\mathcal{E}(P_{\mathbf{p}})$ of a parametric polytope $P_{\mathbf{p}}$ (or a parametric standard $\mathbb{Z}$-polytope $\mathcal{Z} = P_{\mathbf{p}} \cap \mathbb{Z}^d$) is a function from the set of n-dimensional integer vectors $\mathbb{Z}^n$ to the set of natural numbers $\mathbb{N}$. The function value at $\mathbf{p}_0$, denoted $\mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0)$, is the number of integer points in polytope $P_{\mathbf{p}}$:*

$$\begin{array}{rcccc} \mathcal{E}(P_{\mathbf{p}}) & : & \mathbb{Z}^n & \rightarrow & \mathbb{N} \\ & & \mathbf{p}_0 & \mapsto & \mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0) \end{array}$$

$$\mathcal{E}(P_{\mathbf{p}}; \mathbf{p}_0) = \#\left(\left\{\, \mathbf{x} \in \mathbb{Z}^d \mid A\mathbf{x} \geq B\mathbf{p}_0 + \mathbf{c} \,\right\}\right)$$

where # denotes the number of elements of the set.

We first consider a special case of parametric polytopes $P_{\mathbf{p}}$ that can be written as the convex combination of a fixed set of parametric vertices, where each vertex is an affine function of the parameters, i.e.,

$$P_{\mathbf{p}} = \left\{ \mathbf{x} \in \mathbb{Q}^d \mid \mathbf{x} = V(\mathbf{p})\boldsymbol{\lambda}, 0 \leq \lambda_j, \sum_j \lambda_j = 1 \right\}, \tag{6}$$

where the columns $V_j(\mathbf{p})$ of $V(\mathbf{p})$ are the vertices of $P_{\mathbf{p}}$, i.e., $V_j(\mathbf{p}) = G_j\mathbf{p} + \mathbf{h}_j$ for some $G_j \in \mathbb{Q}^{d \times n}$ and $\mathbf{h}_j \in \mathbb{Q}^d$. Ehrhart [11] showed that the enumerator of

such a set can be represented as a *quasi-polynomial* function of $\mathbf{p}$, as defined by (5). That is to say, a polynomial which coefficients depend periodically on $\mathbf{p}$.

Clauss and Loechner [7] showed that the parameter space of a general parametric polytope $P_{\mathbf{p}}$ as defined by (2) can be divided into a set of chambers (called "validity domain" in their paper: a union of adjacent polytopes covering the parameter space), such that in each chamber, $P_{\mathbf{p}}$ has a fixed set of parametric vertices that are affine combinations of the parameters. In each chamber, the parametric polytope can therefore be written as (6). Using this decomposition, Clauss and Loechner [7] proved the following theorem.

Theorem 2.13 *The enumerator of a d-dimensional parametric polytope $P_{\mathbf{p}}$ is a quasi-polynomial (commonly known as Ehrhart quasi-polynomial) of degree d in* $\mathbf{p}$ *on each chamber. The componentwise period of the quasi-polynomial in a given chamber divides the componentwise* lcm *of the denominators that appear in the vertices defined on that chamber.*

We recently proposed a polynomial algorithm (for fixed dimension) [35, 36] that counts lattice points in general parametric polytopes. This algorithm combines Barvinok's counting method [4] with Clauss and Loechner's chamber decomposition [7]. We also use this algorithm to count points in integer affine transformations of parametric $\mathbb{Z}$-polytopes.

3 Integer affine transformations of parametric $\mathbb{Z}$-polytopes

This section presents our algorithm for computing the integer affine transformation of a parametric $\mathbb{Z}$-polytope as a union of $\mathbb{Z}$-polytopes. The calculation of the number of points in such a union is described in Section 4.

It has been shown that the integer affine transformation of a $\mathbb{Z}$-polytope can be written as a Presburger formula [26, 18]:

$$S = \{\mathbf{x} \in \mathbb{Z}^d \mid \exists \mathbf{x}' \in \mathbb{Z}^{d'} : \ A\mathbf{x}' + B\mathbf{x} + C\mathbf{p} + \mathbf{c} = 0, \ A'\mathbf{x}' + B'\mathbf{x} + C'\mathbf{p} + \mathbf{c}' \geq 0\} \tag{7}$$

where A, B, C, A', B', C' are integer matrices, $\mathbf{c}$ and $\mathbf{c}'$ are integer vectors and $\mathbf{p}$ is a parameter vector.

Example 1 *The transformation of the parametric ($N \in \mathbb{Z}$ is a parameter) standard $\mathbb{Z}$-polytope*

$$\mathcal{Z}_N = \{(i,j,k) \in \mathbb{Z}^3 \mid 1 \leq i \leq N \ \wedge 1 \leq j \leq N \ \wedge 1 \leq k \leq N\}$$

by the integer affine function $T(i,j,k) = (3i+6k, 5i+2j+1)$ can be written by adding two variables $x = 3i+6k$ and $y = 5i+2j+1$, as the Presburger formula:

$$S = \{(x,y) \in \mathbb{Z}^2 \mid \exists (i,j,k) \in \mathbb{Z}^3 : \\ 1 \leq i \leq N \ \wedge \ 1 \leq j \leq N \ \wedge \ 1 \leq k \leq N \ \wedge \ x = 3i+6k \ \wedge \ y = 5i+2j+1\}.$$

The problem of calculating the integer affine transformation of a $\mathbb{Z}$-polytope reduces to the elimination of the existential variables from a Presburger formula (Eq. (7)). In such a formula, the conjunction of the equalities and inequalities define a non-full-dimensional standard $\mathbb{Z}$-polytope $\hat{\mathcal{Z}}$ in the combined space $\mathbb{Z}^{d'+d}$ (the cartesian product of the $\mathbb{Z}$-polytope space $\mathbb{Z}^d$ and the existential variables space $\mathbb{Z}^{d'}$). A common way to eliminate the existential variables of the Presburger formula is to project them out of $\hat{\mathcal{Z}}$. Our algorithm performs this in two steps. It starts by eliminating a first set of existential variables using equalities, as described in the next section. Then, our algorithm eliminates the remaining existential variables using inequalities of $\hat{\mathcal{Z}}$.

3.1 Non-full-dimensional $\mathbb{Z}$-polytope preprocessing

When removing existential variables using a set of equalities in $\mathbb{Z}$-polytope $\hat{\mathcal{Z}}$, one must ensure that there exist integer values of the variables to be eliminated for each integer value of the other variables. Without loss of generality, we consider the parameters as regular variables. Indeed, removing equalities from a *parametric* $\mathbb{Z}$-polytope is done in the same way as for a non-parametric $\mathbb{Z}$-polytope: parameters are free variables, and they are never eliminated whatever the number of equalities. We also assume that the $\mathbb{Z}$-polytope is *standard*, since any arbitrary $\mathbb{Z}$-polytope $\mathcal{Z} = P \cap L$ can be transformed into a standard $\mathbb{Z}$-polytope containing the same number of integer points. This is done by computing the preimage of polytope P by a matrix defining lattice L. Notice that it is possible to rewrite the final result as a function of the original variables by doing the inverse transformation, as explained in section 4.

Consider a $(d'+d)$-dimensional standard $\mathbb{Z}$-polytope $\hat{\mathcal{Z}}$, defined by a non-redundant system of linear constraints, with d' existential variables and d free variables. Let m be the number of equalities implying existential variables. The system of equalities is upper-triangularized, with the existential variables on the first columns. Let us denote the set of equalities implying existential variables by $E(\mathbf{y}', \mathbf{y})$, and the set of remaining constraints by $\overline{E}(\mathbf{y}', \mathbf{y})$, where $\mathbf{y}'$ is a vector of m variables chosen among the d' existential variables, and $\mathbf{y}$ is a vector of the k remaining variables, with $k = d' + d - m$.

In order to eliminate the m equalities, it suffices to solve the system of equalities $E(\mathbf{y}', \mathbf{y})$ in $\mathbf{y}'$ as a function of $\mathbf{y}$, and to substitute the result in the system of remaining constraints $\overline{E}(\mathbf{y}', \mathbf{y})$. The resulting standard $\mathbb{Z}$-polytope $\mathcal{Y}$ has then to be intersected with an integer lattice defining the valid values of $\mathbf{y}$. We call this lattice the *validity lattice* of $\mathbb{Z}$-polytope $\mathcal{Y}$. The result is a possibly *non-standard* $\mathbb{Z}$-polytope $\mathcal{Z}' = \mathcal{Y} \cap L$. In the following, we explain how this validity lattice L is calculated.

The system of equalities $E(\mathbf{y}', \mathbf{y})$ can be written:

$$A\mathbf{y}' + B\mathbf{y} + \mathbf{c} = 0, \tag{8}$$

where A is a full row-rank $(m \times m)$ integer matrix, B is an $(m \times k)$ integer matrix and $\mathbf{c}$ is an m-vector. We want to derive the valid values of $\mathbf{y}$ from

equation (8). In other words, we want a necessary and sufficient condition on $\mathbf{y}$ for an integer solution $\mathbf{y}'$ to exist.

Lemma 3.1 *The integer values of* $\mathbf{y}$ *for which the system of equalities (8) admits an integer solution in* $\mathbf{y}'$*, if it exists, are given by a* k*-dimensional integer lattice* $\mathcal{L}(G, \mathbf{y}_0)$*. We call* $\mathcal{L}(G, \mathbf{y}_0)$ *the* validity lattice *over* $\mathbf{y}$ *induced by (8).*

Theorem 1 *The proof of lemma (3.1) is based on the lattice intersection computation [28]. Indeed, the necessary and sufficient condition on* $\mathbf{y}$ *for the existence of an integer solution in* $\mathbf{y}'$ *to (8) is the following:*

$$\exists \mathbf{z} \in \mathbb{Z}^m,\ A\mathbf{y}' = -B\mathbf{y} - \mathbf{c} = \mathbf{z}.$$

or equivalently that there is an integer solution in $(\mathbf{y}', \mathbf{y}, \mathbf{z})$ *to the system of equalities:*

$$\begin{pmatrix} -A & 0 & I_m \\ 0 & B & I_m \end{pmatrix} \begin{pmatrix} \mathbf{y}' \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{c} \end{pmatrix}, \tag{9}$$

where I_m *is the* $(m \times m)$ *identity matrix.*

The left hand-side matrix in equation (9) is, by construction, a $(2m \times (2m + k))$ *full row-rank integer matrix. Hence, the integer solution to (9), if it exists, is given as a function of* k *free variables:*

$$\begin{pmatrix} \mathbf{y}' \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} E \\ G \\ F \end{pmatrix} \mathbf{t} + \begin{pmatrix} \mathbf{y}'_0 \\ \mathbf{y}_0 \\ \mathbf{z}_0 \end{pmatrix},\ \mathbf{t} \in \mathbb{Z}^k. \tag{10}$$

The condition on $\mathbf{y}$ *for the existence of an integer solution to (10) is then:*

$$\mathbf{y} = G\mathbf{t} + \mathbf{y}_0,\ \mathbf{t} \in \mathbb{Z}^k,$$

where G *is a* $(k \times k)$ *integer matrix and* $\mathbf{y}_0$ *is an integer* k*-vector. Hence,* $\mathbf{y}$ *should be part of the* k*-dimensional lattice* $\mathcal{L}(G, \mathbf{y}_0)$*.*

Example 2 *Consider the Presburger formula:*

$$\begin{aligned} S \quad = \{(x, y, z) \in \mathbb{Z}^3 \mid \exists (i, j) \in \mathbb{Z}^2 :\ & 1 \leq i \leq N \wedge 1 \leq j \leq N \wedge 1 \leq x \leq N \wedge \\ & 3i + 6j = y \wedge 5i = -2x + z - 1\}. \end{aligned} \tag{11}$$

In order to eliminate the equalities of this formula it suffices to eliminate variables i *and* j *(as in the rational case), and to intersect the result with the validity lattice corresponding to values of* x*,* y*,* z*, and* N *for which the system of equalities admits an integer solution for* (i, j)*. The rational elimination of the equalities results in a full-dimensional standard* $\mathbb{Z}$*-polytope:*

$$\mathcal{Y} = \{(x, y, z) \in \mathbb{Z}^3 \mid 27 \leq 6x + 5y - 3z \leq 30N - 3 \wedge 6 \leq -2x + z \leq 5N + 1 \wedge 1 \leq x \leq N\}.$$

However, not all integer values of (x, y, z) in this set correspond to integer values of (i, j). The validity lattice of this $\mathbb{Z}$-polytope is obtained by solving the system of equalities:

$$\left(\begin{array}{cc|cccc|cc} -3 & -6 & 0 & 0 & 0 & 0 & 1 & 0 \\ -5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & -1 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} i \\ j \\ x \\ y \\ z \\ N \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

The solution of this system can be computed as:

$$\begin{pmatrix} i \\ j \\ x \\ y \\ z \\ N \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 5 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Hence,

$$\begin{pmatrix} x \\ y \\ z \\ N \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

which defines an integer lattice $\mathcal{L}(G, \mathbf{y}_0) = \mathcal{L}\left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 \\ 2 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}\right)$.

The final result of eliminating equalities of formula (11) is the new $\mathbb{Z}$-polytope $\mathcal{Z}' = \mathcal{Y} \cap \mathcal{L}(G, \mathbf{y}_0)$.

To continue the computation, let us call $\mathcal{Z}' = \mathcal{Y} \cap L$ the resulting $\mathbb{Z}$-polytope after eliminating the equalities implying existential variables in a $\mathbb{Z}$-polytope $\hat{\mathcal{Z}}$. Two cases can occur.

- There are no remaining existential variables (it occurs when the number of existential variables is smaller or equal to the number of equalities that involve them, as in the example presented above). Then, $\mathcal{Z}'$ is the solution.
- There still remain existential variables to be eliminated in $\mathcal{Z}'$. In this case, we first reorganize the lattice so that its basis matrix becomes upper triangular. We then calculate the compression of $\mathbb{Z}$-polytope $\mathcal{Z}'$, *i.e.*, the

preimage of $\mathcal{Y}$ by the matrix defining lattice L. Finally, we eliminate the remaining existential variables as explained in section 3.3. In order to preserve the original coordinates, we calculate the transformation of the resulting $\mathbb{Z}$-polytopes by the submatrix obtained by removing the lines and columns corresponding to existential variables from the basis matrix of L. The result is then intersected with the sublattice defined by this latter submatrix to obtain the desired transformation.

In the following, we consider without loss of generality, that our formulas do not contain equalities implying existential variables, and we focus on eliminating the remaining existential variables.

3.2 Existential variable elimination and integer projection

A classical algorithm to eliminate variables from a system of constraints is the Fourier-Motzkin elimination procedure [9]. It allows the elimination of a *rational* existential variable from a system of affine inequalities, defined on the set of rational numbers. Its main idea consists in rewriting the original system as a set of lower and upper bounds on the variable to be eliminated. Then, each pair of lower and upper bounds of the form $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$ is to be replaced by $\alpha l(\mathbf{x},\mathbf{p}) \leq \beta u(\mathbf{x},\mathbf{p})$, where z is the existential variable chosen to be eliminated, $l(\mathbf{x},\mathbf{p})$ and $u(\mathbf{x},\mathbf{p})$ are affine functions of variables $\mathbf{x}$ and parameters $\mathbf{p}$ independent of z, and α and β are strictly positive integer constants. This procedure has been extended to integer existential variable elimination by W. Pugh et al. [25, 26, 27] as follows:

Any pair of lower and upper bounds $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$ defines:

- an **exact shadow**, corresponding to the *rational* projection of the points belonging to this pair of constraints. This is given by $\alpha l(\mathbf{x},\mathbf{p}) \leq \beta u(\mathbf{x},\mathbf{p})$.
- a **dark shadow**, corresponding to the convex part of the exact shadow in which any integer point has at least one integer preimage. This is given by $\alpha l(\mathbf{x},\mathbf{p}) + (\alpha-1)(\beta-1) \leq \beta u(\mathbf{x},\mathbf{p})$. Notice that if $\alpha = 1$ or $\beta = 1$, the dark shadow is equal to the exact shadow.

The part of the exact shadow that does not belong to the dark shadow may contain integer points having integer preimages, and other integer points called *holes* having only rational preimages in $P_{\mathbf{p}}$ (see Figure 1).

The Omega test [25] answers the question: "is there an integer point in the projection having at least an integer preimage?" as follows:

- if the exact shadow does not contain any integer point, the answer is *no*,
- if the dark shadow contains at least one integer point, the answer is *yes*,
- otherwise, the answer is not obvious. In this case, we have to know whether the part of the exact shadow which does not belong to the dark shadow contains an integer point having integer preimage(s) in $P_{\mathbf{p}}$.

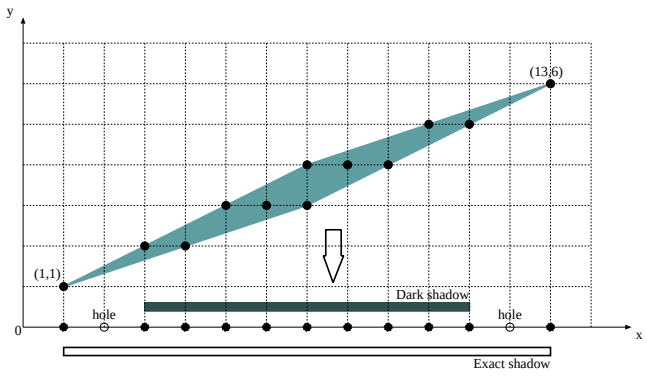

Figure 1: The integer projection of a $\mathbb{Z}$-polytope.

In order to answer this latter question, Pugh and Wonnacott [27] check whether the intersection between a certain number (function of α and β) of hyperplanes and the constraints of the original system contains an integer point (see section 5).

Note 1 *The integer projection of a $\mathbb{Z}$-polytope results in a* unique *dark shadow (whatever the number of existential variables) and a union of $\mathbb{Z}$-polytopes. Hence, the challenge is that of computing such a union.*

3.3 Our projection method

In this section, we focus on the projection of a single pair of lower and upper bounds of the existential variable chosen to be eliminated. Indeed, the projection of the whole $\mathbb{Z}$-polytope is obtained by intersecting the projections of all its pairs of bounds (like the well-known Fourier-Motzkin algorithm). The result is also intersected with the constraints that are independent of the eliminated variable.

Consider a pair of lower and upper bounds $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$. The projection of such a pair is given by the union of its dark shadow $(\alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) + (\alpha-1)(\beta-1) \leq 0)$ and another set of integer points which can not be obtained by applying simple rules. The following theorem defines the hyperplanes on which the integer points lie.

Lemma 3.2 *Let x, y be two rational numbers and $\lceil x \rceil$, $\lceil y \rceil$ (resp. $\lfloor x \rfloor$, $\lfloor y \rfloor$) be their upper (resp. lower) integer parts. The following properties are equivalent:*

1. $\exists n \in \mathbb{Z}$ *such that* $x \leq n \leq y$,
2. $\lceil x \rceil \leq \lfloor y \rfloor$,
3. $\lceil x \rceil \leq y$,
4. $x \leq \lfloor y \rfloor$.

Theorem 3.3 *Consider the pair of bounds* $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$ *and let* $l(\mathbf{x},\mathbf{p}) = l_l(\mathbf{x},\mathbf{p}) + c_l$ *and* $u(\mathbf{x},\mathbf{p}) = l_u(\mathbf{x},\mathbf{p}) + c_u$*, where* $l_l(\mathbf{x},\mathbf{p})$ *and* $l_u(\mathbf{x},\mathbf{p})$ *are linear functions,* c_l *and* c_u *are integer constants. Let* g *be the greatest common divisor (gcd) of the coefficients of variables* $\mathbf{x}$ *and parameters* $\mathbf{p}$ *in the linear function* $(\alpha l_l(\mathbf{x},\mathbf{p}) - \beta l_u(\mathbf{x},\mathbf{p}))$*. Then*

- *the points outside the dark shadow which belong to the integer projection lie on hyperplanes of the form:*

$$\alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) + \gamma = 0, \tag{12}$$

  with $\gamma \in \mathbb{Z}$*,* $0 \leq \gamma \leq \alpha\beta - \alpha - \beta$ *and* g *divides* $(\beta c_u - \alpha c_l - \gamma)$*.*

- *the values of* γ *for which the hyperplane (12) contains the points we are interested in are those verifying the following inequality:*

$$\alpha(-l(\mathbf{x},\mathbf{p}) \bmod \beta) \leq \gamma, \tag{13}$$

  which is equivalent to:

$$\beta(u(\mathbf{x},\mathbf{p}) \bmod \alpha) \leq \gamma. \tag{14}$$

Theorem 2 *We recall that the exact and the dark shadows are respectively given by* $\alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) \leq 0$ *and* $\alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) \leq -(\alpha-1)(\beta-1)$ *[27]. By definition, the part of the exact shadow containing the points outside the dark shadow, which belong to the projection (see Figure 1), is given by* $-(\alpha\beta-\alpha-\beta) \leq \alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) \leq 0$*. This is equivalent to* $\alpha l(\mathbf{x},\mathbf{p}) - \beta u(\mathbf{x},\mathbf{p}) + \gamma = 0$*, where* γ *is an integer constant such that* $0 \leq \gamma \leq \alpha\beta - \alpha - \beta$*.*

By substituting the values of $l(\mathbf{x},\mathbf{p}) = l_l(\mathbf{x},\mathbf{p}) + c_l$ *and* $u(\mathbf{x},\mathbf{p}) = l_u(\mathbf{x},\mathbf{p}) + c_u$ *in (12) we obtain* $\alpha l_l(\mathbf{x},\mathbf{p}) - \beta l_u(\mathbf{x},\mathbf{p}) = \beta c_u - \alpha c_l - \gamma$*, where* $(\alpha l_l(\mathbf{x},\mathbf{p}) - \beta l_u(\mathbf{x},\mathbf{p}))$ *is a linear function and* $(\beta c_u - \alpha c_l - \gamma)$ *is an integer constant. A necessary and sufficient condition for this hyperplane to contain integer points is that the gcd of the coefficients in the linear function* $(\alpha l_l(\mathbf{x},\mathbf{p}) - \beta l_u(\mathbf{x},\mathbf{p}))$ *divides the constant* $(\beta c_u - \alpha c_l - \gamma)$*.*

On the other hand, $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$ *is equivalent to* $\frac{l(\mathbf{x},\mathbf{p})}{\beta} \leq z \leq \frac{u(\mathbf{x},\mathbf{p})}{\alpha}$ *(since* $\alpha, \beta > 0$*), where* $\frac{l(\mathbf{x},\mathbf{p})}{\beta}$ *and* $\frac{u(\mathbf{x},\mathbf{p})}{\alpha}$ *are rational functions. According to properties 1 and 2 of Lemma 3.2, there exists an integer* z *such that* $\frac{l(\mathbf{x},\mathbf{p})}{\beta} \leq z \leq \frac{u(\mathbf{x},\mathbf{p})}{\alpha}$ *if and only if:*

$$\left\lceil \frac{l(\mathbf{x},\mathbf{p})}{\beta} \right\rceil \leq \frac{u(\mathbf{x},\mathbf{p})}{\alpha}, \tag{15}$$

with $\left\lceil \frac{l(\mathbf{x},\mathbf{p})}{\beta} \right\rceil = \frac{1}{\beta}(l(\mathbf{x},\mathbf{p}) + (-l(\mathbf{x},\mathbf{p})) \bmod \beta)$*. By simplifying (15), we obtain:* $\alpha(-l(\mathbf{x},\mathbf{p}) \bmod \beta) \leq \beta u(\mathbf{x},\mathbf{p}) - \alpha l(\mathbf{x},\mathbf{p})$*, with* $\beta u(\mathbf{x},\mathbf{p}) - \alpha l(\mathbf{x},\mathbf{p}) = \gamma$ *(according to equality (12)). Hence, inequality (13) is satisfied. Applying Lemma 3.2, we can similarly prove inequality (14).*

Note 2 *If one of the bounds $l(\mathbf{x},\mathbf{p})$ or $u(\mathbf{x},\mathbf{p})$ is independent of the variables and the parameters, the elimination of the existential variable results in only one convex region, because in this case, $\left\lceil \frac{l(\mathbf{x},\mathbf{p})}{\beta} \right\rceil$ or $\left\lfloor \frac{u(\mathbf{x},\mathbf{p})}{\alpha} \right\rfloor$ is a constant.*

Example 3 *Consider the pair of bounds $\{7 \leq 4z,\ 3z \leq 2x+5p+1\}$. That is to say, $l(\mathbf{x},\mathbf{p}) = 7$ (independant of the variables and parameters), $u(\mathbf{x},\mathbf{p}) = 2x+5p+1$, $\alpha = 3$ and $\beta = 4$. The integer elimination of z results, applying inequality (15), in only one constraint: $\left\lceil \frac{7}{4} \right\rceil \leq \frac{2x+5p+1}{3}$, which is equivalent to $2x+5p+1 \geq 5$.*

In the following, we describe how to calculate the solutions of inequalities (13) or (14), in two different ways, depending on whether the coefficients α and β are coprime or not.

3.3.1 Case of coprime coefficients

Theorem 3.4 *Consider the pair of bounds $\{l(\mathbf{x},\mathbf{p}) \leq \beta z,\ \alpha z \leq u(\mathbf{x},\mathbf{p})\}$, with α, β coprime. The calculation of the values of γ for which hyperplane (12) contains the points, outside the dark shadow, and that belong to the integer projection does not depend on the variables and the parameters, i.e., it depends only on the constants α and β. In this case, inequalities (13) and (14) are respectively equivalent to (16) and (17).*

$$\alpha((c_1\gamma) \bmod \beta) \leq \gamma, \tag{16}$$

$$\beta((c_2\gamma) \bmod \alpha) \leq \gamma, \tag{17}$$

where c_1 and c_2 are integer constants such that $c_1\alpha + c_2\beta = 1$ (computing these constants is straightforward from Bezout's identity theorem).

Theorem 3 *Since α and β are coprime, there exists two integer constants c_1 and c_2 such that (Bezout's identity theorem):*

$$c_1\alpha + c_2\beta = 1. \tag{18}$$

Multiplying equality (12) by c_1, we obtain $c_1\alpha l(\mathbf{x},\mathbf{p}) - c_1\beta u(\mathbf{x},\mathbf{p}) + c_1\gamma = 0$. This is equivalent to $(1-c_2\beta)l(\mathbf{x},\mathbf{p}) - c_1\beta u(\mathbf{x},\mathbf{p}) + c_1\gamma = 0$ (according to (18)). Hence,

$$l(\mathbf{x},\mathbf{p}) = \beta(c_2 l(\mathbf{x},\mathbf{p}) + c_1 u(\mathbf{x},\mathbf{p})) - c_1\gamma.$$

Substituting the value of $l(\mathbf{x},\mathbf{p})$ in inequality (13), we obtain:

$$\alpha((-\beta(c_2 l(\mathbf{x},\mathbf{p}) + c_1 u(\mathbf{x},\mathbf{p})) + c_1\gamma) \bmod \beta) \leq \gamma \Leftrightarrow \alpha(c_1\gamma \bmod \beta) \leq \gamma.$$

This proves Eq. (16). In the same way, one can prove (17), starting from inequality (14).

Example 4 *Consider the following Presburger formula:*

$$\mathcal{S} = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} :\ 2 \leq 3y - x \leq 5 \wedge -1 \leq x - 2y \leq N - 1\},$$

where N is a positive integer parameter. This set is equivalent to the projection on x of the $\mathbb{Z}$-polytope $\mathcal{Z}$ pictured in Figure 1 for $N = 2$.

According to the pair of bounds $\{x - N + 1 \leq 2y,\ 3y \leq x + 5\}$, we have:

$$l(x, N) = x - N + 1,\ u(x, N) = x + 5,\ \alpha = 3,\ \beta = 2.$$

We can choose $c_1 = 1$ and $c_2 = -1$ ($c_1\alpha + c_2\beta = 1$).

The constraint on the dark shadow corresponding to this pair of bounds is $x \leq 3N + 5$. The points of the projection that do not belong to the dark shadow are given by:

$$\alpha l(x, N) - \beta u(x, N) + \gamma = 0,\ 0 \leq \gamma \leq \alpha\beta - \alpha - \beta \text{ and } \alpha((c_1\gamma) \bmod \beta) \leq \gamma$$

$$\Rightarrow x - 3N - 7 + \gamma = 0,\ 0 \leq \gamma \leq 1 \text{ and } 3(\gamma \bmod 2) \leq \gamma.$$

Scanning the values of γ, we find that the only one satisfying these constraints is $\gamma = 0$. The corresponding hyperplane is $x = 3N + 7$. Similarly, one can calculate the point $x = 1$ from the other pair of bounds $\{x + 2 \leq 3y,\ 2y \leq x + 1\}$ generating the constraint $x \geq 3$ on the dark shadow.

The integer projection of $\mathbb{Z}$-polytope $\mathcal{Z}$ is then obtained by intersecting the projections of both pairs, i.e., $S = \{x = 3N + 7 \vee x \leq 3N + 5\} \cap \{x = 1 \vee x \geq 3\}$. Since N is positive, this set is equal to:

$$\mathcal{S} = \{x \in \mathbb{Z} \mid x = 1 \vee 3 \leq x \leq 3N + 5 \vee x = 3N + 7\}.$$

3.3.2 Case of non-coprime coefficients

In the previous subsection, we showed how we calculate the projection of a pair of bounds when the coefficients α and β are coprime. Let us now consider the case of non-coprime coefficients. In this case, the calculation of the values of γ for which the hyperplane (12) contains the points of the projection, and that are outside the dark shadow, depends not only on α and β, but also on the variables and the parameters. Let $g' = gcd(\alpha, \beta)$, $\alpha' = \alpha/g'$, $\beta' = \beta/g'$ and g be the *gcd* of the coefficients of the variables and parameters in the linear function $\alpha' l_l(\mathbf{x}, \mathbf{p}) - \beta' l_u(\mathbf{x}, \mathbf{p})$, with $l(\mathbf{x}, \mathbf{p}) = l_l(\mathbf{x}, \mathbf{p}) + c_l$ and $u(\mathbf{x}, \mathbf{p}) = l_u(\mathbf{x}, \mathbf{p}) + c_u$ (see Theorem 3.3). One can then rewrite the equation of the hyperplane (12) as follows:

$$\alpha' l(\mathbf{x}, \mathbf{p}) - \beta' u(\mathbf{x}, \mathbf{p}) + \gamma = 0, \tag{19}$$

with $\gamma \in \mathbb{Z}$, $0 \leq \gamma \leq \alpha\beta' - \alpha' - \beta'$ and g divides $(\beta' c_u - \alpha' c_l - \gamma)$.
Inequalities (13) and (14) can be respectively rewritten as (20) and (21):

$$\alpha'(-l(\mathbf{x}, \mathbf{p}) \bmod \beta) \leq \gamma, \tag{20}$$

$$\beta'(u(\mathbf{x},\mathbf{p}) \bmod \alpha) \leq \gamma. \tag{21}$$

In this case, it may happen that only a subset of the integer points of hyperplane (19) belongs to the projection. These points are defined by the intersection between the hyperplane and a union of lattices obtained by solving one of the following equalities.

$$-l(\mathbf{x},\mathbf{p}) \bmod \beta = \gamma', \text{ with } 0 \leq \gamma' \leq \min\left(\left\lfloor \frac{\gamma}{\alpha'} \right\rfloor, \beta\right), \tag{22}$$

$$u(\mathbf{x},\mathbf{p}) \bmod \alpha = \gamma', \text{ with } 0 \leq \gamma' \leq \min\left(\left\lfloor \frac{\gamma}{\beta'} \right\rfloor, \alpha\right). \tag{23}$$

In practice, it is worth considering equality (22) when $\beta < \alpha$ and equality (23) otherwise.

The solution to a *modulo* equality $f(\mathbf{x},\mathbf{p}) \bmod a = b$ is a lattice of the form:

$$L = \left\{ \left(\begin{array}{c|c} A_{\mathbf{x}} & A_{\mathbf{p}} \end{array} \right) \left(\begin{array}{c} \mathbf{x} \\ \mathbf{p} \end{array} \right) + \mathbf{c} \;\middle|\; \mathbf{x} \in \mathbb{Z}^d,\ \mathbf{p} \in \mathbb{Z}^n \right\}, \tag{24}$$

where $A_{\mathbf{x}}, A_{\mathbf{p}}$ are integer matrices, $\mathbf{c}$ is an integer vector, $\mathbf{x}$ is a vector of the variables space and $\mathbf{p}$ is a vector of parameters. We calculate this solution using the technique presented in section 3.1. Indeed, $f(\mathbf{x},\mathbf{p}) \bmod a = b$ is equivalent to $\exists z \in \mathbb{Z} : f(\mathbf{x},\mathbf{p}) = az + b$. It then suffices to calculate the validity lattice of this latter equation by eliminating existential variable z, or in other words, all integer values of $\mathbf{x}$ and $\mathbf{p}$ for which variable z is integer. Of course, only non-empty lattices and hyperplanes are taken into account.

Example 5 *Consider a pair of bounds for which the coefficients of the existential variable y are not coprime $\{x - N - 2 \leq 2y,\ 4y \leq x + 5\}$. We have:*

$$l(x,N) = x - N - 2,\ u(x,N) = x + 5,\ \alpha = 4,\ \beta = 2$$

$$\Rightarrow gcd(\alpha,\beta) = 2,\ \alpha' = 2,\ \beta' = 1.$$

The corresponding constraint on the dark shadow is $2x \leq 4N+15 \Leftrightarrow x \leq 2N+7$. The points outside the dark shadow and belonging to the projection lie on the following hyperplane:

$$x - 2N - 9 + \gamma = 0, \text{ such that } 0 \leq \gamma \leq 1 \text{ and } 2((x - N - 2) \bmod 2) \leq \gamma.$$

For both values of γ, the solution to the above modulo *inequality is a lattice:*

$$L = \left\{ \left(\begin{array}{cc} 2 & 1 \\ 0 & 1 \end{array} \right) \left(\begin{array}{c} x \\ N \end{array} \right) + \left(\begin{array}{c} -2 \\ 0 \end{array} \right) \;\middle|\; x \in \mathbb{Z},\ N \in \mathbb{Z} \right\}.$$

Hence, points $x = 2N + 9$ and $x = 2N + 8$ (obtained by substituting the values of γ in equality $x - 2N - 9 + \gamma = 0$) belong to the projection only if x and N belong to lattice L.

The whole projection of the pair of bounds is then given by:

$$\mathcal{S} = \{x \in \mathbb{Z} \mid (x = 2N+8 \wedge (x,N) \in L) \vee (x = 2N+9 \wedge (x,N) \in L) \vee x \leq 2N+7\}.$$

Algorithm 1 Calculating the integer affine transformation of a $\mathbb{Z}$-polytope

Input:
$\mathcal{Z}$: $\mathbb{Z}$-polytope
T: Transformation matrix
Output:
$\mathcal{Z}_u$: Union of $\mathbb{Z}$-polytopes
Variables:
F, U: Union of $\mathbb{Z}$-polytopes

F = EliminateEqualities (PresburgerFormula $(\mathcal{Z}, T)$)
// Remaining existential variables elimination
For each z in variables to be eliminated
  F = ReduceLattice (z, F)
  U = Universe (Dim$(F) - 1$) // $U = \mathbb{Z}^{Dim(F)-1}$
  For each $(\alpha u, \beta l)$ in pairs of bounds on z in F
    D = DarkShadow $(\alpha u, \beta l)$
    If $\alpha = 1$ or $\beta = 1$
      $U = U \cap D$
    Else
      E = ExactShadow $(\alpha u, \beta l)$
      $U = U \cap (D \cup RemoveHoles(E - D, \alpha u, \beta l))$
    End If
  End For
  $F = U$
End For
$\mathcal{Z}_u = F$

Projecting out a first existential variable from a $\mathbb{Z}$-polytope may result in a union of *non-standard* $\mathbb{Z}$-polytopes. Therefore, in order to project out a second variable, this union has to be projected again, and so on. The projection of each non-standard $\mathbb{Z}$-polytope is obtained by first transforming it into a standard $\mathbb{Z}$-polytope, then projecting it as explained before. The result is finally rewritten as a function of the original variables and parameters. The whole method of calculating the integer affine transformation of a $\mathbb{Z}$-polytope is summarized in Algorithm 1.

The result is given as a union of parametric $\mathbb{Z}$-polytopes. In the following section, we will be interested in counting points in such unions.

4 Counting points in unions of $\mathbb{Z}$-polytopes

We will now discuss an algorithm that deals with unions of parametric $\mathbb{Z}$-polytopes of the form $\mathcal{Z} = P \cap L$, with:

$$P = \left\{ \mathbf{x} \in \mathbb{Q}^d, \mathbf{p} \in \mathbb{Z}^n \;\middle|\; \left(\begin{array}{c|c} A_{\mathbf{x}} & A_{\mathbf{p}} \end{array} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \mathbf{a} \geq 0 \right\},$$
$$L = \left\{ \left(\begin{array}{c|c} B_{\mathbf{x}} & B_{\mathbf{p}} \end{array} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} + \mathbf{b} \;\middle|\; \mathbf{x} \in \mathbb{Z}^d,\ \mathbf{p} \in \mathbb{Z}^n \right\},$$

where P is a parametric polytope, L is a parametric integer lattice, $A_{\mathbf{x}}, A_{\mathbf{p}}, B_{\mathbf{x}}$ and $B_{\mathbf{p}}$ are integer matrices, $\mathbf{a}$ and $\mathbf{b}$ are integer vectors, $\mathbf{x}$ is a vector of the variables space and $\mathbf{p}$ is a vector of parameters.

Let $\mathcal{Z}_1 = P_1 \cap L_1$ and $\mathcal{Z}_2 = P_2 \cap L_2$ be two $\mathbb{Z}$-polytopes. The number of points of $\mathcal{Z}_1$ is equal to the number of integer points contained in P_1', where P_1' is the transformation of P_1 by the matrix B_1 defining lattice L_1. In the same way, the number of points of $\mathcal{Z}_2$ is equal to the number of integer points in $P_2' = B_2 P_2$. Unfortunately, the number of points in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is obviously not equal to that in $P_1' \cup P_2'$ since the applied transformations preserve the number of points in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ but do not preserve their original coordinates.

Example 6 *Consider the two $\mathbb{Z}$-polytopes pictured in Figure 2, $\mathcal{Z}_1 = P_1 \cap L_1$ and $\mathcal{Z}_2 = P_2 \cap L_2$, where dots belong to $\mathcal{Z}_1$, squares belong to $\mathcal{Z}_2$ and diamonds belong to both $\mathbb{Z}$-polytopes:*

$$P_1 = \{(x, y) \in \mathbb{Q}^2 \mid 1 \leq x \leq 10 \ \wedge \ 3 \leq y \leq 7\},$$

$$L_1 = \left\{ \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \;\middle|\; (x', y') \in \mathbb{Z}^2 \right\},$$

$$P_2 = \{(x, y) \in \mathbb{Q}^2 \mid 3 \leq x \leq 12 \ \wedge \ 1 \leq y \leq 6\},$$

$$L_2 = \left\{ \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \;\middle|\; (x', y') \in \mathbb{Z}^2 \right\}.$$

Substituting $x = 2x' + 1$ and $y = 2y' + 1$ in P_1 (resp. $x = 3x'$ and $y = 2y' + 1$ in P_2) we obtain P_1' and P_2' in which the numbers of points are respectively 15 *and* 12*:*

$$P_1' = \{(x', y') \in \mathbb{Z}^2 \mid 0 \leq 2x' \leq 9 \ \wedge \ 1 \leq y' \leq 3\},$$

$$P_2' = \{(x', y') \in \mathbb{Z}^2 \mid 1 \leq x' \leq 4 \ \wedge \ 0 \leq 2y' \leq 5\}.$$

One can check that the number of points in $P_1' \cup P_2'$ is 19*, whereas that in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is* 23 *as shown in Figure 2.*

The previous methods [22, 39] to count points in unions of $\mathbb{Z}$-polytopes are *lattice-union* based, which is exponential in the size of lattice generators and their least common multiple. Furthermore, Zhu et al.'s method [39] only deals with *non-parametric* $\mathbb{Z}$-polytopes.

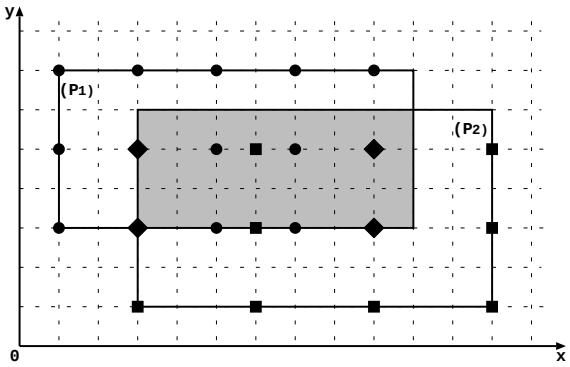

Figure 2: Union of two $\mathbb{Z}$-polytopes

In contrast, our method is *lattice-intersection* based: its complexity is polynomial, since the intersection between two lattices results in only one lattice, whatever their generators. Previous algorithms start by calculating a *disjoint* union of the input $\mathbb{Z}$-polytopes. It is usually very hard to separate a union of $\mathbb{Z}$-polytopes into a *disjoint* union [39], and its complexity may be exponential even for a fixed number of $\mathbb{Z}$-polytopes.

In our algorithm (Algorithm 2), we start by applying the inclusion-exclusion principle to the union of $\mathbb{Z}$-polytopes. We therefore process on a set of signed $\mathbb{Z}$-polytopes: $\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2)$, the number of integer points in the union of two $\mathbb{Z}$-polytopes $\mathcal{Z}_1$ and $\mathcal{Z}_2$, is equal to $\mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_1 \cap \mathcal{Z}_2)$.

After applying the inclusion-exclusion principle, the number of points in each resulting $\mathbb{Z}$-polytope $\mathcal{Y}_i = P_i \cap L_i$ is calculated as follows:

We first transform the matrix generating the lattice: $L_i = \left\{ \left(\begin{array}{c|c} B_{\mathbf{x}} & B_{\mathbf{p}} \end{array} \right) \left(\begin{array}{c} \mathbf{x} \\ \mathbf{p} \end{array} \right) + \left(\begin{array}{c} \mathbf{b_x} \\ \mathbf{b_p} \end{array} \right) \;\middle|\; \mathbf{x} \in \mathbb{Z}^d, \mathbf{p} \in \mathbb{Z}^n \right\}$ into a new matrix of the form: $M = \left(\begin{array}{c|c} B_{\mathbf{xx}} & B_{\mathbf{xp}} \\ \hline 0 & B_{\mathbf{pp}} \end{array} \right)$. Matrix M generates the same integer points as the original matrix, and the rows of M defining the lattice in the parameter space are *independent* of the variables. This is required to keep the variables space compressed when rewriting the transformed $\mathbb{Z}$-polytope as a function of its original parameters. We calculate matrix M from the Hermite normal form [30] of matrix $\left(\begin{array}{c|c} B_{\mathbf{x}} & B_{\mathbf{p}} \end{array} \right)$, with rows and columns exchanged such that parameters appear first. This is needed to obtain zeroes under $B_{\mathbf{xx}}$ in M. We apply the affine transformation $\left(\begin{array}{c|c} M & \left(\begin{array}{c} \mathbf{b_x} \\ \mathbf{b_p} \end{array} \right) \end{array} \right)$ to P_i to get a polytope P'. Then, P' is rewritten as a function of the original parameters using the submatrix $(B_{\mathbf{pp}} \mid \mathbf{b_p})$ and finally, we use our counting algorithm [35, 36] to calculate the Ehrhart quasi-polynomial corresponding to the number of integer points in the resulting polytope. Note that when submatrix $B_{\mathbf{pp}}$ is not equal to the identity matrix, the polytope is valid for only the parameter values generated by lattice

Algorithm 2 Counting integer points in a union of parametric $\mathbb{Z}$-polytopes

Input:
F: Union of $\mathbb{Z}$-polytopes
Output:
L: List of (Validity domain, Ehrhart quasi-polynomial)
Variables:
S, S': List of (sign, $\mathbb{Z}$-polytope)
P: Polytope; I: $\mathbb{Z}$-polytope

// Inclusion-exclusion principle
S = Empty
For each $\mathcal{Z}$ in F
  $S' = S$
  For each (s,$\mathcal{Y}$) in S
    $I = \mathcal{Z} \cap \mathcal{Y}$
    If Not Empty (I)
      $S' = S' + (-1 \times s, I)$
    End If
  End For
  $S = S' + (+1, \mathcal{Z})$
End For

// Enumeration of S
L= Empty ()
For each $(s, \mathcal{Y})$ in S
  $\mathcal{Y}.L$ = NormalizeLattice($\mathcal{Y}.L$)
  P = Transform($\mathcal{Y}.P, \mathcal{Y}.L$),
  L = AddAndSimplify ($L, s\times$Enumerate (P))
End For

$L_{\mathbf{p}}$ whose basis is $B_{\mathbf{pp}}$ and affine part is $\mathbf{b_p}$. In this case, the resulting Ehrhart quasi-polynomial is to be multiplied by one if the parameter values are valid (i.e., if they belong to the points spanned by lattice $L_{\mathbf{p}}$) and zero otherwise.

Example 7 *Consider parametric versions of $\mathbb{Z}$-polytopes $\mathcal{Z}_1 = P_1 \cap L_1$ and $\mathcal{Z}_2 = P_2 \cap L_2$ from Example 6, with*

$$P_1 = \{(x, y) \in \mathbb{Q}^2 \mid 1 \leq x \leq N + 5 \ \wedge \ 3 \leq y \leq 7\},$$

$$L_1 = \left\{ \left(\begin{array}{cc|c} 2 & 0 & 3 \\ 0 & 2 & 0 \\ \hline 0 & 0 & 1 \end{array} \right) \left(\begin{array}{c} x' \\ y' \\ N' \end{array} \right) + \left(\begin{array}{c} 1 \\ 1 \\ 0 \end{array} \right) \ \middle| \ (x', y') \in \mathbb{Z}^2, \ N' \in \mathbb{Z} \right\},$$

$$P_2 = \{(x,y) \in \mathbb{Q}^2 \mid 3 \leq x \leq 2N+7 \ \wedge \ 1 \leq y \leq 6\},$$

$$L_2 = \left\{ \left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ \hline 0 & 0 & 3 \end{array} \right) \left(\begin{array}{c} x' \\ y' \\ N' \end{array} \right) + \left(\begin{array}{c} 2 \\ 1 \\ 0 \end{array} \right) \ \middle| \ (x',y') \in \mathbb{Z}^2, \ N' \in \mathbb{Z} \right\},$$

where $N \in \mathbb{Z}^+$ is a parameter. The number of points in the union $\mathcal{Z}_1 \cup \mathcal{Z}_2$ is given by:

$$\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2) = \mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_1 \cap \mathcal{Z}_2).$$

Let $(\mathcal{Z}_1 \cap \mathcal{Z}_2) = \mathcal{Z}_3 = (P_3 \cap L_3)$, with

$$P_3 = P_1 \cap P_2 = \{(x,y) \in \mathbb{Q}^2 \mid 3 \leq x \leq N+5 \ \wedge \ 3 \leq y \leq 6\},$$

$$L_3 = L_1 \cap L_2 = \left\{ \left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ \hline 3 & 0 & 6 \end{array} \right) \left(\begin{array}{c} x' \\ y' \\ N' \end{array} \right) + \left(\begin{array}{c} 2 \\ 1 \\ 3 \end{array} \right) \ \middle| \ (x',y') \in \mathbb{Z}^2, \ N' \in \mathbb{Z} \right\}.$$

The number of integer points in $\mathcal{Z}_3$ is calculated as follows.

First of all, basis $\left(\begin{array}{cc|c} 3 & 0 & 0 \\ 0 & 2 & 0 \\ \hline 3 & 0 & 6 \end{array} \right)$ of lattice L_3 is transformed into a new basis $M = \left(\begin{array}{cc|c} 6 & 0 & 3 \\ 0 & 2 & 0 \\ \hline 0 & 0 & 3 \end{array} \right)$ in which the variable coefficients in the parameter row are all equal to zero. The new basis spans the same integer points as the original one since it is calculated from its Hermite normal form. $\mathbb{Z}$-polytope $\mathcal{Z}_3$ is then transformed into a regular polytope P given by the preimage of P_3 by homogeneous matrix $\left(\begin{array}{ccc|c} 6 & 0 & 3 & 2 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 3 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$:

$$P = \{(x',y') \in \mathbb{Q}^2 \mid -3N'+1 \leq 6x' \leq 6 \ \wedge \ 2 \leq 2y' \leq 5\}.$$

Before counting points in P, we need to write it as a function of the original parameter N. To do this, it suffices to calculate its transformation by matrix $M = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 3 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$, and we obtain:

$$P' = \{(x',y') \in \mathbb{Q}^2 \mid -N+4 \leq 6x' \leq 6 \ \wedge \ 2 \leq 2y' \leq 5\}.$$

The number of integer points in this polytope is given (using the algorithm described in [35, 36]) by:

$$\mathcal{E}(P') = \frac{1}{3}N + \left[2, \frac{5}{3}, \frac{4}{3}, 1, \frac{8}{3}, \frac{7}{3}\right]_N,$$

where $\left[2, \frac{5}{3}, \frac{4}{3}, 1, \frac{8}{3}, \frac{7}{3}\right]_N$ *is a periodic number whose value is* 2 *when* $N \bmod 6 = 0$, $\frac{5}{3}$ *when* $N \bmod 6 = 1$ *and so on.*

The third row in matrix M *states that* $N = 3N' + 3$*, with* $N' \in \mathbb{Z}$*. In other words,* N *must be a multiple of* 3 *(*$N \bmod 3 = 0$*). Therefore the resulting polynomial is multiplied by a periodic number* $[1, 0, 0]_N$ *which is equal to one when* $N \bmod 3 = 0$ *and zero otherwise. The result is then:*

$$\mathcal{E}(\mathcal{Z}_3) = \left[\frac{1}{3}, 0, 0\right]_N N + [2, 0, 0, 1, 0, 0]_N$$

The numbers of points in $\mathcal{Z}_1$ *and* $\mathcal{Z}_2$ *are obtained in a similar way as:*

$$\mathcal{E}(\mathcal{Z}_1) = \frac{3}{2}N + \left[9, \frac{15}{2}\right]_N,$$

$$\mathcal{E}(\mathcal{Z}_2) = [2, 0, 0]_N N + [3, 0, 0]_N.$$

Finally, the number of points in $\mathcal{Z}_1 \cup \mathcal{Z}_2$ *is:*

$$\mathcal{E}(\mathcal{Z}_1 \cup \mathcal{Z}_2) = \mathcal{E}(\mathcal{Z}_1) + \mathcal{E}(\mathcal{Z}_2) - \mathcal{E}(\mathcal{Z}_3) = \left[\frac{19}{6}, \frac{3}{2}, \frac{3}{2}\right]_N N + \left[10, \frac{15}{2}, 9, \frac{19}{2}, 9, \frac{15}{2}\right]_N.$$

The complexity of our algorithm (Algorithm 2) depends on the complexity of the significant polytope and $\mathbb{Z}$-polytope operations, the complexity of counting integer points in a parametric polytope and the number of the resulting $\mathbb{Z}$-polytopes after applying the inclusion-exclusion principle:

- The polytope operations rely on PolyLib, which is based on the dual *rays* and *constraints* representations. All the operations that we use, namely transform and intersection, are polynomial for fixed dimension.
- The only significant $\mathbb{Z}$-polytope operation used in this algorithm is the intersection, which is polynomial since the intersection between two polytopes is a concatenation of their constraints, and the intersection between two lattices reduces to solving a system of linear equalities [23], which is polynomial in the input size [30].
- Counting integer points in a parametric polytope is also polynomial in the input size (for fixed dimension), as we showed in [35, 36].
- Finally, the $\mathbb{Z}$-polytopes to be enumerated are given by the inclusion-exclusion principle. When the number of input $\mathbb{Z}$-polytopes is fixed, the inclusion-exclusion principle provides a polynomial number of (non-empty) $\mathbb{Z}$-polytopes.

Hence, the whole algorithm is polynomial in the input size (for fixed dimension and fixed number of input $\mathbb{Z}$-polytopes).

5 Related work and applications

In this section, we compare our work to some recent and interesting methods over a set of problems that frequently arise in code optimization techniques. All the execution times that are given were measured on a 3GHz Intel Pentium 4 with 1GB of RAM running Linux 2.6.23.

5.1 Omega test

We first illustrate the difference between our method and Pugh et al.'s Omega test [25, 26, 27]. The Omega test answers the question: *is there an integer point in the integer projection of a polytope?* as follows: it first computes the *exact shadow* (real projection). If it does not contain any integer point, then the answer is "no". Then it computes the *dark shadow*. If it contains at least an integer point, then the answer is "yes"; else Pugh and Wonnacott [27] propose to check whether the intersection between a certain number of hyperplanes and the constraints of the original system contains an integer point. Their solution provides new constraints with possibly extra existential variables (the *splinters*) which increases the complexity of further computations, such as:

- counting the number of integer points contained in the integer projection of the $\mathbb{Z}$-polytope?
- projecting the result along another dimension?

Note that the splinters provided by Pugh's method are somewhat similar to our $\mathbb{Z}$-polytopes (when the coefficients of the existential variable are not coprime, see Section 3.3.2). But it is not clear whether these splinters can be projected along another dimension without scanning a possibly exponentially growing number of $\mathbb{Z}$-polytopes. Also, when the coefficients are coprime, Pugh's method does not propose a simple solution as we do, and no explanation is given for non-coprime coefficients nor for the parametric case.

Example 8 *Consider the following example (introduced in [27]):*

$$\mathcal{S} = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} :\ 0 \leq 3y - x \leq 7 \wedge 1 \leq x - 2y \leq 5\}.$$

The exact shadow defined by the elimination of y is given by $3 \leq x \leq 29$ and the dark shadow by $5 \leq x \leq 27$.

Pugh and Wonnacott [27] calculate the set of constraints containing the images which do not belong to the dark shadow as follows:

$$\begin{array}{l} \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} :\ x = 3y \wedge 1 \leq y \leq 5\ \} \cup \\ \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} :\ x = 3y - 1 \wedge 2 \leq y \leq 6\ \} \cup \\ \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z} :\ x = 2y + 5 \wedge 5 \leq y \leq 12\} \end{array}$$

which has to be intersected with the difference between the exact shadow and the dark shadow in order to obtain the solution. While our rules provide these images directly as $\{x = 3,\ x = 29\}$.

5.2 Weak quantifier elimination

A recent and interesting work manipulating Presburger formulas is the one due to Lasaruk and Sturm [19]. It consists in a generalization of Presburger arithmetic, where the coefficients are arbitrary polynomials in non-quantified variables. The authors use the term "weak" to refer to the fact that the result may still contain bounded quantifiers. In order to expand this result into disjunctions, the parameters have to be instantiated. In the following, we will show that, at least for a subset of problems[1], our method also handles parameters and can do better, in particular in terms of execution time.

Let us consider an interesting issue raised by the code optimization community, which is dependency analysis for automatic parallelization. More precisely, we consider the example from [19] (section 5.2). The goal is to check whether a data dependency occurs or not in the following loop nest:

```
for i = 0 to m do
   for j = 0 to m do
      A[n*i+j] = i+j
```

A data dependency for example occurs when two or more different loop iterations write different values to a same array element. For this loop nest, this condition can be expressed as the following Presburger formula:

$$\exists i \exists j \exists i' \exists j' (0 \leq i \leq m \ \wedge \ 0 \leq j \leq m \ \wedge \ 0 \leq i' \leq m \ \wedge \ 0 \leq j' \leq m \wedge \\ (i \neq i' \ \vee \ j \neq j') \ \wedge \ i + j \neq i' + j' \ \wedge \ ni + j = ni' + j').$$

Lasaruk and Sturm [19] compute in $3.53s$ that the weak quantifier elimination over integers from the above formula results in 25441 atomic formulas, which can not be expanded unless the parameters are substituted. They also raise the fact that it is more efficient to first plug in values for the parameters m and n, and then to perform quantifier elimination plus expansion. When they do so, they obtain "true" for $m = n = 4$ in $0.28s$ and "false" for $m = 4$ and $n = 5$ in $0.29s$.

Let us now see how this problem can efficiently be solved using our method:

When $m = n = 4$ we obtain "yes" in $0.01s$, and we obtain "false" when $m = 4$ and $n = 5$ in less than $0.01s$.

Better, our method has only to plug in the value of parameter n (since we do not handle polynomial constraints). In this case, we obtain:

- when $n = 4$: "true" if $m >= 4$ and "false" otherwise in less than $0.01s$,
- when $n = 5$: "true" if $m >= 5$ and "false" otherwise in $0.01s$.

Notice that the authors do not mention what could be the result in this second case.

[1] Our method handles only linear constraints (polynomial constraints are not supported).

Another information that can be given by our method is the array elements causing the data dependency and their count. This is done by replacing equality $ni + j = ni' + j'$ by $x = ni + j \wedge x = ni' + j'$ in the previous formula, where x is a free-quantifier variable representing accessed array elements. Again, only parameter n has to be substituted. For example, when $n = 4$, the number of array cells causing a dependency is given as: $\{4$ if $m = 4, 10$ if $m = 5, 18$ if $m = 6$ and $5m - 7$ if $m \geq 7\}$.

5.3 Barvinok-Woods's integer projection

In 2003, Barvinok and Woods [3] proposed an algorithm for computing the integer projection of a polytope that is *theoretically* polynomial-time for fixed dimension. However, no implementation of this algorithm has been reported till 2008. Because it contains complicated theory such as algorithmic flatness theory and iterated Boolean combinations of rational generating functions, this algorithm has frequently been referred to as "unimplementable" [17]. In 2008, Köppe et al. [17] rose to the challenge of proposing the first-ever implementation of Barvinok-Woods's algorithm. In their paper, the authors present the results of preliminary computational experiments. Unfortunately, their implementation turned out to be rather inefficient, as shown in the experiments they reported, in particular *woods_2.1.7* and *scarf_1*. The following table summarizes execution times (in seconds) obtained by Köppe et al.'s implementation.

	ex1	*woods_2.1.7*	*pugh*	*scarf_1*
# parameters	0	1	0	2
# free-quantifier variables	0	0	0	0
# existentially quantified variables	2	2	2	2
# inequalities	4	4	4	5
Köppe's time without exploiting small gapes in dimension 2	0.11	29.2	0.09	126
Köppe's time exploiting small gapes in dimension 2	0.08	2.7		1.1

For all these examples, our method computes the result in less than $0.01s$. Notice that we did not test our method on the *param.pugh* example, because the authors did not make it public.

5.4 Verdoolaege's integer projection

In this section, we compare our implementation with the one written by Sven Verdoolaege and distributed in the Barvinok library [33]. This implementation consists in applying simple rewriting rules to a set of disjoint union of parametric sets computed using the Omega library [16]. If these rules fail, the PIP library [12] is used to solve a parametric integer linear programming problem, in the way of Boulet and Redon [6]. Notice that both tools (Omega and PIP) are worst-case exponential.

The particular advantage of our method over the one of Verdoolaege is its ability to handle efficiently *unions* of projections. Indeed, in such a case, our algorithm computes each projection independently of the others, even if the original sets are not pairwise disjoint projections. The final result is then easily obtained by doing the union of the resulting atomic projections. The number of lattice points contained in this union is computed straightforwardly using the algorithm proposed in section 4. Verdoolaege's method could not do so since, in contrast to our method, it does not compute the actual atomic projections but equivalent sets having the same number of lattice points. Hence, the original input sets have to be pairwise disjoint. Furthermore, the equivalent resulting sets are usually of higher dimension than the actual projections. This, usually, leads to higher lattice points counting times and results in larger Ehrhart quasi-polynomials.

Example 9 *Consider the following two loop nests accessing a one-dimensional array:*

```
for i = 1 to n do
   for j = i+1 to n do
      A[2*i+3*j] = ...

for k = 1 to n do
   for l = 1 to k-1 do
      A[k+2*l] = ...
```

Suppose we are interested in the number of array elements that are reached by at least one of these loop nests iterations. This is useful, for example, to compute the amount of data being accessed by this piece of program (the cache size used by this computation could also be computed). This in turn reduces to the problem of counting the solutions to the following Presburger formula:

$$\exists i \exists j (1 \leq i \leq n \ \wedge \ i+1 \leq j \leq n \ \wedge \ x = 2i + 3j) \vee$$
$$\exists k \exists l (1 \leq k \leq n \ \wedge \ 1 \leq l \leq k-1 \ \wedge \ x = k + 2l).$$

*Verdoolaege's algorithm first uses the Omega library to compute three disjoint formulas which still contain existential variables, and then applies the rewriting rules and/or the PIP library to compute their integer projections, which results in a set of disjoint polytopes. Applying the lattice points counting algorithm to this set results in 3-degree Ehrhart quasi-polynomials. The solution is given in 0.12*s *computation time and the size of the output file is 2475 bytes.*

*In contrast to this, our method computes the result in only 0.01*s *and the output file takes 256 bytes only. The result we obtain is:* $2, 6, 11, 16, 22$ *when* n *respectively goes from* 2 *to* 6*, and* $5n - 7$ *when* n *is greater or equal to 7. Notice that the size of the output file has to be as small as possible, since it contains Ehrhart quasi-polynomials that are sometimes used in optimizing code applications.*

In order to measure the efficiency of our method, we also ran a simulation through a thousand of pseudo-randomly generated examples that model array accesses in perfect loop nests. The examples are constructed as follows:

- the depth of loop nests (number of the existential variables) varies from 1 to 6,
- the dimension of arrays (number of free-quantifier variables) varies from 1 to 4,
- the number of parameters varies from 1 to 4,
- the number of equalities equals the array dimension,
- the number of inequalities equals 2 times the depth of the loop nest,
- the coefficients of the variables and parameters are chosen such that they reflect what could be found in a real program.

Consequently, the number of dimensions of the generated sets[2] varies from 3 to 14, and the number of constraints varies from 3 to 16. For these 1000 examples, we get the following execution times:

- For 804 examples, our implementation and Verdoolaege's compute the solution in $0.01s$ or less, and for 7 examples the two implementations have the same computation time (between $0.02s$ and $0.10s$).
- For 94 examples, our implementation does better than Verdoolaege's.
- For 46 examples, Verdoolaege's implementation does better than ours.
- For 49 examples, both implementations do not compute the solution in less than $30s$, which we set as timeout threshold.

Notice that, even if all the generated examples in this simulation are single sets (no unions of projections have been computed), our method yields to better results than the one of Verdoolaege.

This simulation was undertaken with the PolyLib library version 5.23 and the Barvinok library version 0.28. In both implementations, PolyLib is used to realize polyhedral operations, and the Barvinok library to count integer points in parametric polytopes. In addition, Verdoolaege uses the PIP library [12] and the Omega library [16] to simplify the input polytopes. All the computations have been performed on a 3GHz Intel Pentium 4 with 1GB of RAM.

[2] We consider only non-empty sets.

6 Conclusion

We presented a new algorithm for calculating the integer affine transformation of parametric $\mathbb{Z}$-polytopes. The solution is given as a union of parametric $\mathbb{Z}$-polytopes, worst-case exponential but efficient in practical cases, and less complex compared to other existing methods. A general polynomial algorithm for computing the exact solution remains a challenge. We also proposed a new polynomial algorithm to count points in arbitrary unions of a fixed number of $\mathbb{Z}$-polytopes (of fixed dimension).

These algorithms have been implemented using the PolyLib library [20] and the Barvinok library [35, 36]. They have many applications, particularly in compiler design (for example analyzing and optimizing loop nests), and in other domains such as economics and mathematics (for example in combinatorics, representation theory, statistics, discrete optimization).

References

[1] M. Welleda Baldoni-Silva, Matthias Beck, Charles Cochet, and Michèle Vergne. Volume computation for polytopes and partition functions for classical root systems. *Discrete & Computational Geometry*, 35(4):551–595, 2006.

[2] A. I. Barvinok. Computing the Ehrhart polynomial of a convex lattice polytope. *Discrete Comput. Geom.*, 12:35–48, 1994.

[3] Alexander Barvinok and Kevin Woods. Short rational generating functions for lattice point problems. *Journal of the American Mathematical Society*, 16:657–979, 2003.

[4] Alexander I. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19(4):769–779, 1994.

[5] B. Boigelot and L. Latour. Counting the solutions of Presburger equations without enumerating them. *Theoretical Computer Science*, 313(1):17–29, Feb. 2004.

[6] Pierre Boulet and Xavier Redon. Communication pre-evaluation in HPF. In *EUROPAR'98*, volume 1470 of *LNCS*, pages 263–272. Springer Verlag, 1998.

[7] Philippe Clauss and Vincent Loechner. Parametric Analysis of Polyhedral Iteration Spaces. *Journal of VLSI Signal Processing*, 19(2):179–194, July 1998.

[8] P. D'Alberto, A. Veidembaum, A. Nicolau, and R. Gupta. Static analysis of parameterized loop nests for energy efficient use of data caches. In

Workshop on Compilers and Operating Systems for Low Power (COLP01), September 2001.

[9] George B. Dantzig and B. Curtis Eaves. Fourier-Motzkin elimination and its dual. *J. Comb. Theory, Ser. A*, 14(3):288–297, 1973.

[10] Jesús A. De Loera, Raymond Hemmecke, Jeremiah Tauzer, and Ruriko Yoshida. Effective lattice point counting in rational convex polytopes. *Journal of Symbolic Computation*, 38(4):1273–1302, 2004.

[11] E. Ehrhart. Polynômes arithmétiques et méthode des polyèdres en combinatoire. *International Series of Numerical Mathematics*, 35, 1977.

[12] P. Feautrier, J. Collard, and C. Bastoul. Solving systems of affine (in)equalities. Technical report, PRiSM, Versailles University, 2002.

[13] Paul Feautrier. Parametric integer programming. *Recherche Operationnelle/Operations Research*, 22(3):243–268, 1988.

[14] Somnath Ghosh, Margaret Martonosi, and Sharad Malik. Cache miss equations: a compiler framework for analyzing and tuning memory behavior. *ACM Transactions on Programming Languages and Systems*, 21(4):703–746, 1999.

[15] Felix Heine and Adrian Slowik. Volume driven data distribution for NUMA-machines. In *Proceedings from the 6th International Euro-Par Conference on Parallel Processing*, pages 415–424, 2000.

[16] W. Kelly, V. Maslov, W. Pugh, E. Rosser, T. Shpeisman, and D. Wonnacott. The Omega Library. Technical report, Institue for advanced computer studies, University of Maryland, College Park, 1996.

[17] Matthias Köppe, Sven Verdoolaege, and Kevin Woods. An implementation of the barvinok–woods integer projection algorithm. In *ITSL*, pages 53–59, 2008.

[18] G. Kreisel and J. L. Krevine. *Elements of Mathematical Logic*. The Netherlands: North-Holland Publishing, 1967.

[19] Aless Lasaruk and Thomas Sturm. Weak quantifier elimination for the full linear theory of the integers: A uniform generalization of presburger arithmetic. *Appl. Algebra Eng., Commun. Comput.*, 18(6):545–574, 2007.

[20] Vincent Loechner. Polylib: A library for manipulating parameterized polyhedra. Technical report, LSIIT - ICPS UMR7005 Univ. Louis Pasteur-CNRS, 1999.

[21] Vincent Loechner, Benoît Meister, and Philippe Clauss. Precise data locality optimization of nested loops. *Journal of Supercomputing*, 21(1):37–76, 2002.

[22] B. Meister. Projecting periodic polyhedra for loop nest analysis. In *Proceedings of the 11th Workshop on Compilers for Parallel Computers (CPC 04), Kloster Seeon, Germany*, pages 13–24, July 2004.

[23] S P K. Nookala and T. Risset. A library for Z-polyhedral operations. Technical report, 1330, Irisa, 2000.

[24] Erin Parker and Siddhartha Chatterjee. An automata-theoretic algorithm for counting solutions to Presburger formulas. In *Compiler Construction 2004*, volume 2985 of *Lecture Notes in Computer Science*, pages 104–119, April 2004.

[25] William Pugh. The Omega test: a fast and practical integer programming algorithm for dependence analysis. In *Proceedings of the 1991 ACM/IEEE conference on Supercomputing*, pages 4–13. ACM Press, 1991.

[26] William Pugh. Counting solutions to Presburger formulas: how and why. In *SIGPLAN Conference on Programming Language Design and Implementation (PLDI'94)*, pages 121–134, 1994.

[27] William Pugh and David Wonnacott. Experiences with constraint-based array dependence analysis. In *Principles and Practice of Constraint Programming*, pages 312–325, 1994.

[28] Patrice Quinton, Sanjay Rajopadhye, and Tanguy Risset. On manipulating Z-polyhedra using a canonical representation. *Parallel Processing Letters*, 7(2):181–194, 1997.

[29] J. Ramanujam, Jinpyo Hong, Mahmut Kandemir, and A. Narayan. Reducing memory requirements of nested loops for embedded systems. In *DAC '01: Proceedings of the 38th conference on Design automation*, pages 359–364, New York, NY, USA, 2001. ACM Press.

[30] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1986.

[31] Rachid Seghir and Vincent Loechner. Memory optimization by counting points in integer transformations of parametric polytopes. In *In Proceedings of International Conference on Compilers, Architectures, and Synthesis for Embedded Systems, CASES 2006*, pages 74–82, October 2006.

[32] Alexandru Turjan, Bart Kienhuis, and Ed Deprettere. Solving out-of-order communication in Kahn process networks. *J. VLSI Signal Process. Syst.*, 40(1):7–18, 2005.

[33] Sven Verdoolaege. Barvinok: user guide, August 2006. http://www.kotnet.org/~skimo/barvinok/.

[34] Sven Verdoolaege, Kristof Beyls, Maurice Bruynooghe, and Francky Catthoor. Experiences with enumeration of integer projections of parametric polytopes. In R. Bodik, editor, *Compiler Construction: 14th International Conference*, volume 3443, pages 91–105, Edinburgh, March 2005. Springer.

[35] Sven Verdoolaege, Rachid Seghir, Kristof Beyls, Vincent Loechner, and Maurice Bruynooghe. Analytical computation of Ehrhart polynomials: Enabling more compiler analyses and optimizations. In *Proceedings of International Conference on Compilers, Architectures, and Synthesis for Embedded Systems, Washington D.C.*, pages 248–258, September 2004.

[36] Sven Verdoolaege, Rachid Seghir, Kristof Beyls, Vincent Loechner, and Maurice Bruynooghe. Counting integer points in parametric polytopes using barvinok's rational functions. *Algorithmica*, 48(1):37–66, 2007.

[37] Sven Verdoolaege and Kevin Woods. Counting with rational generating functions, 2005. http://arxiv.org/abs/math/0504059.

[38] Ying Zhao and Sharad Malik. Exact memory size estimation for array computations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(5):517–521, October 2000.

[39] Hongwei Zhu, Ilie I. Luican, and Florin Balasa. Memory size computation for real-time multimedia applications based on polyhedral decomposition. *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences.*, E89-A(12):3378–3386, 2006.